S E N S ' 2 0 0 6
Second Scientific Conference with International Participation
SPACE, ECOLOGY, NANOTECHNOLOGY, SAFETY
14 – 16 June 2006, Varna, Bulgaria

SIMULATION ANALYSIS OF THE VITERBI CONVOLUTIONAL DECODING ALGORITHM

Teodor Iliev

University of Rousse, Department of Communication Technique and Technologies,
e-mail: tediliev@yahoo.com

Keywords: convolutional codes, Viterbi algorithm, Hamming distance;

Abstract. *The advantage of Viterbi decoding, compared with brute–force decoding, is that the complexity of Viterbi decoding is not a function of the number of symbols in the codeword sequence. The Viterbi algorithm removes from consideration those trellis paths that could not possibly be candidates for the maximum likelihood choice. The decoder continues in this way to advance deeper into the trellis, making decision by eliminating the least likely paths. The paper is devoted to an example of Viterbi convolutional decoding, that the goal of selecting the optimum path can be expressed, equivalently, as choosing the codeword with the maximum likelihood metric, or as choosing the codeword with the minimum Hamming distance. We propose encoding and decoding structure with their trellis diagrams and algorithm for hard and soft decoding decision. The received results from the simulation model provide the opportunity of assessing the quality of decoding.*

INTRODUCTION

Convolutional encoding is a powerful method for forward error correction of a binary sequence in digital communications systems. The maximum likelihood (ML) estimation of the information bits gives the best performance as far as the block error rate is concerned [1]. A convolutional code can be represented by a trellis diagram. Starting from a given initial state, a binary sequence determines a unique path in the trellis. The Viterbi algorithm is an efficient way to find the best path in the trellis. For each state in the trellis, the algorithm recursively updates the best path ending in the state, which is called a survivor path. The architecture of a Viterbi decoder consists of three main units: a branch metric computation unit (BMU), an add-compare unit (ACSU) and a trace-back unit (TBU).

Each time there are 2^{K-1} states in the trellis, where K is the constraint length, and each state can be entered by means of two paths. Viterbi decoding consist of computing the metric for the two paths entering each state and eliminating one of them. This computation is done for each of the 2^{K-1} nodes at time t_i, then the decoder moves to time t_{i+1} and repeats the process.[1,3]

Viterbi – Algorithm:

1. For $i = 0,1,...L$ computation of path metric for each path from state a to the other states;
2. $i = i + 1$: computation of the accumulated path metric by adding the branch metric;
3. For each state selection of the path with maximum metric (survivor);
4. Repetition of step 2 and 3 if $i < N + L$;

An example of Viterbi convolutional decoding:

$\underline{X} = (101) \qquad ; n = 1, N = 3$

$\underline{Y} = (11\ 10\ 00\ 10\ 11) \ ; \ m = 2, \ M = 10$

$\downarrow error$

$\underline{Z} = (11\ 11\ 00\ 10\ 11) \qquad$ error in position 4

$p_0 < 0{,}5$ bit error probability

$p(z_\mu = 0 | \hat{y}_\mu = 1) = p(z_\mu = 1 | \hat{y}_\mu = 0) = p_0$

$p(z_\mu = 0 | \hat{y}_\mu = 0) = p(z_\mu = 1 | \hat{y}_\mu = 1) = 1 - p_0$

abreviation:

$\log(1 - p_0) = \alpha$ without errors

$\log(p_0) = \beta$ with errors

$\alpha > \beta$

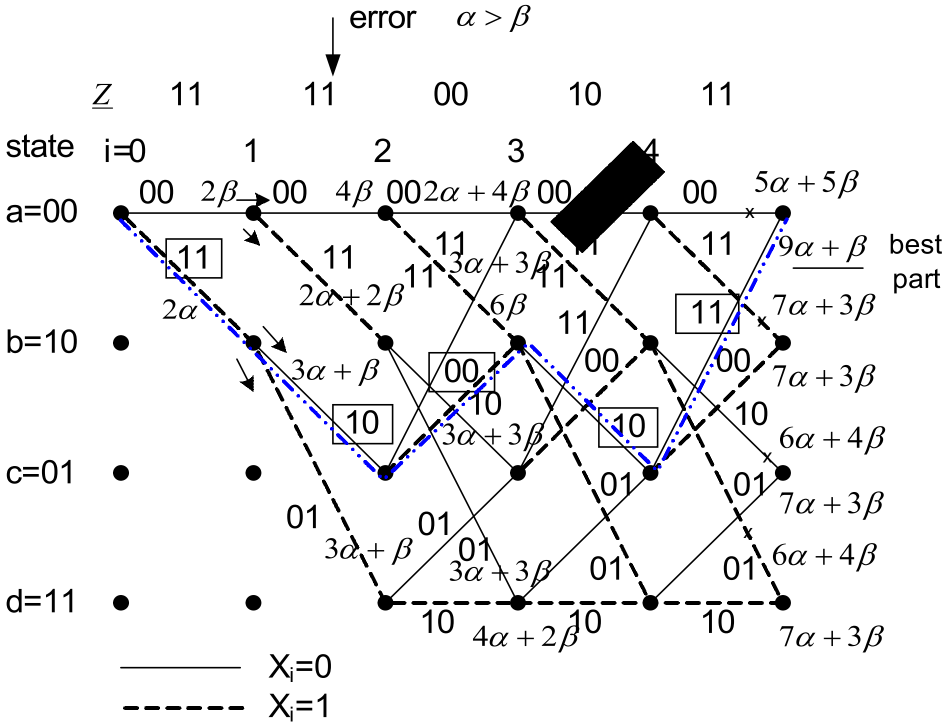


Fig.1: Decoder trellis diagram

$\hat{\underline{Y}} = (11\ 10\ 00\ 10\ 11)$

$\hat{\underline{X}} = (1 \quad 0 \quad 1 \quad 0 \quad 0)$

Viterbi–Algorithm with Hamming–Distance–Metric

For the decoder trellis it is convenient to label each trellis branch at time t_i with the Hamming distance between the received code symbols and corresponding branch word from the encoder trellis. The decoding algorithm uses these Hamming distance metrics to find the most likely (minimum distance) path trough the trellis.[3]

Hamming–Distance:

$d(\underline{Z}, \hat{\underline{Y}}) =$ number of disturbed bits

$M - d(\underline{Z}, \hat{\underline{Y}}) =$ number of undisturbed bits

Transition probabilities:

$$p(z_\lambda|\hat{y}_\lambda)=\begin{cases}p_0 & ;\text{bit error}\\ 1-p_0 & ;\text{error free}\end{cases}$$

$$F=\sum_{\lambda=1}^{M}\log(p(z_\lambda|\hat{y}_\lambda))=d(\underline{Z},\hat{\underline{Y}})\cdot\log(p_0)+(M-d(\underline{Z},\hat{\underline{Y}}))\log(1-p_0)=$$

$$d(\underline{Z},\hat{\underline{Y}})\cdot\underbrace{\log\frac{p_0}{1-p_0}}_{=const}+\underbrace{M\cdot\log(1-p_0)}_{=const}$$

because of $p_0<0{,}5$, $\log(p_0)<0$ and $M\log(p_0)=const<0$

→ maximization of F corresponds to minimization of d.

→ simplified VA with minimization of the Hamming path metric:

$$\widetilde{F}(k)=\sum_{i=1}^{k}\log d(\underline{Z}_i,\hat{\underline{Y}}_i)$$

→ bit error probability must not be known!

Soft–Decision Decoding [4]

Until now we have considered hard – decision decoding according to:

$\underline{Y}=(y_1,y_2...y_M)$ with $y_i\in\{0,1\}$

and

$\underline{Z}=(z_1,z_2...z_M)$ with $z_i\in\{0,1\}$

Real transmission channel are analog.

→ at the output of the demodulator we get analog samples w

→ the elements of $\underline{Z}$ are obtained by quantizing the samples of w, this is called decision.

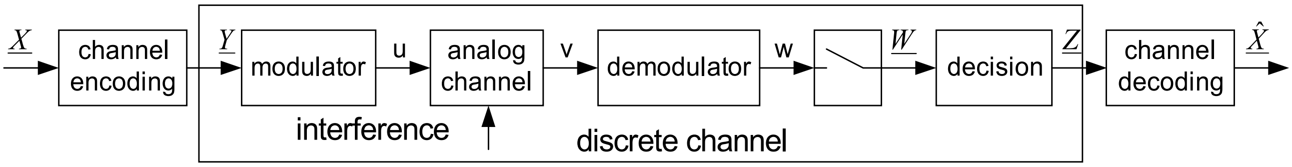


Fig. 2: Transmission model

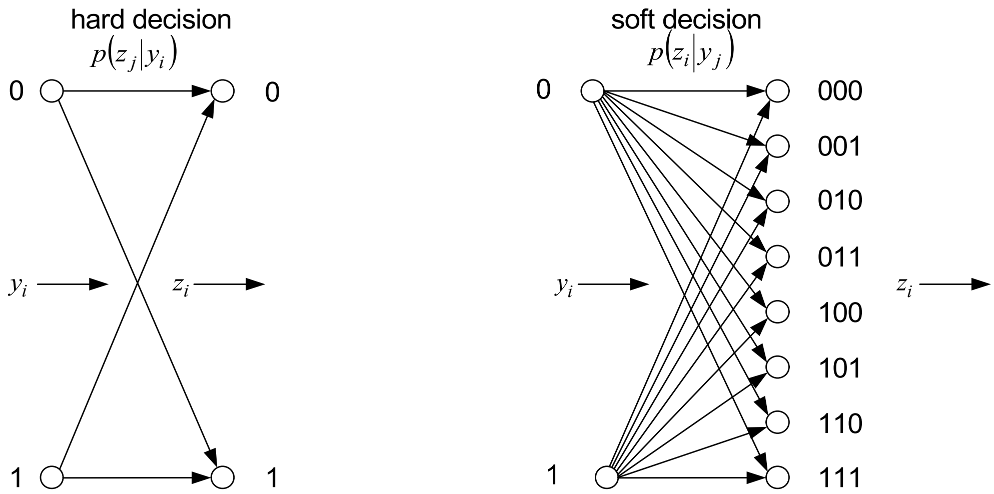


Fig. 3: Hard and soft decision

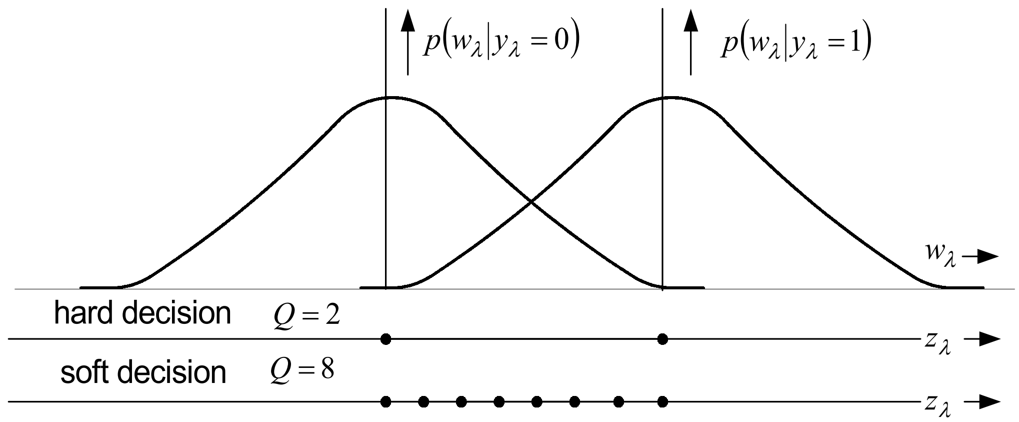


Fig. 4: Hard and soft decoding decision

hard–decision: $z_i \in \{0,1\}$

soft–decision: $0 \le z_i \le 1$, e.g. quantization within the range with 3 bits

w is disturbed by Gaussian noise

→ conditional probability densities

By soft–decision (quantization of w) we get additional information about the reliability of a decision $z_i = 0$ or $z_i = 1$

Conditional probability functions on the premise of an Additive White Gaussian Channel (AWGN).

$$p(w|y_i) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(w-y_i)^2}{2\sigma^2}}$$

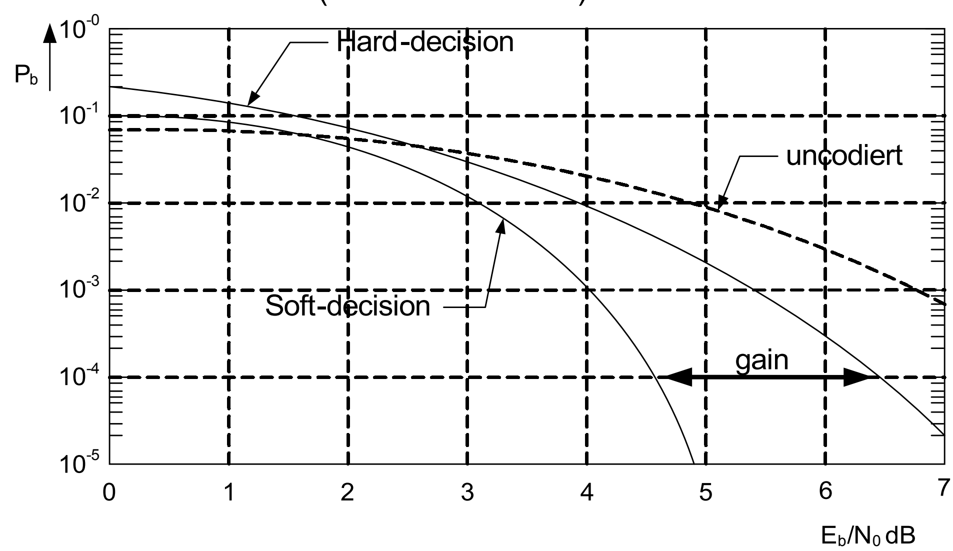
Viterbi – Algoritm with Euklidean Metric

$$\widetilde{F} = \sum_{\lambda=1}^{M} (z_\lambda - \hat{y}_\lambda)^2$$

- Minimization of $\widetilde{F}$
- In case of infinite fine quantization there is a gain of $\Delta SNR \approx 2{,}2dB$
- In case of 8 – level quantization the gain becomes $\Delta SNR \approx 2\,dB$

Example: Comparison of Viterbi–Decoding with Soft– and Hard–Decision.

GSM convolutional encoder (Full Rate channel)



Rate Compatible Punctured Convolutional Codes RCPC)

Modification of coding rate (error protection) by periodic puncturing of the n/m - rate "mother code".

Example: puncturing of a 1/2 – rate code.

Input $\underline{X} = (x_0, x_1, x_2, x_3, x_4) = (11011)$

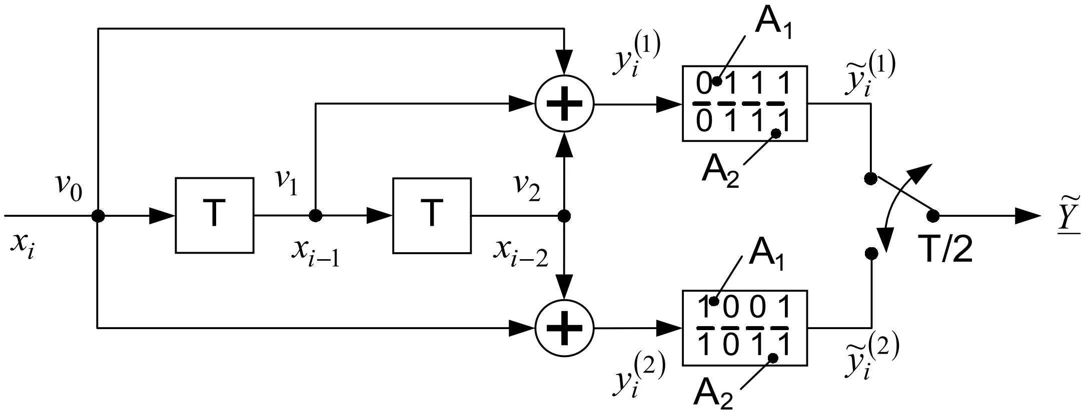


Fig. 5: Block diagram (puncturing with period $p = 4$)

Without puncturing

$$\underline{Y} = \begin{pmatrix} \overset{1}{11} \vdots \overset{2}{01} \vdots \overset{3}{01} \vdots \overset{4}{00} \vdots \overset{1..}{01} \end{pmatrix}$$

With puncturing using $A_1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$

$$\underline{\tilde{Y}} = \left(\underbrace{\_1|0\_|0\_|00|}_{\text{puncturing period}}\_1\ldots \right.$$

4 information symbols are transformed into 5 bits

→ code rate $r = \dfrac{4}{5}$

Puncturing matrix $\underline{A}_1$

$$A_1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

→ 5 out of 8 elements are non – zero →coding rate:

$$\frac{n}{m'} = \frac{n}{m} \cdot \frac{8}{5} = \frac{1}{2} \cdot \frac{8}{5} = \frac{4}{5}$$

Puncturing matrix $\underline{A}_2$

$$A_2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

→ 6 out of 8 elements are non – zero → coding rate:

$$\frac{1}{2} \cdot \frac{8}{6} = \frac{2}{3}$$

CONCLUSIONS

By switching the puncturing scheme, unequal error protection can be achieved according to the different bit error sensitivity of different bit classes of a coded speech frame. The major drawback of the Viterbi algorithm is that while error probability decreases exponentially with constraint length, the number of code states, and consequently decoder complexity, grows exponentially with constraint length. On the other hand, the computational complexity of the Viterbi algorithm is independent of the channel characteristics (compared to hard–decision decoding, soft–decision decoding requires only a trivial increase in the number of computations)

REFERENCES

1. Илиев, Т., Петков, Г., „Цифрова обработка и пренос на сигнали”, Печатна база при РУ „Ангел Кънчев”, 2005 г.
2. Abbasfar, A., Yao, K., „Survivor memory reductions in the Viterbi algorithm”, IEEE Communications Letters, vol. 9, № 4, pp. 352 – 354, April 2005
3. Sklar, B., „Digital Communications“, 2nd ed., Prentice Hall PTR, New Jersey, 2002
4. Vary, P., „Advanced channel coding and modulation“, master program, Communication Engineering in RWTH – Aachen, 2005
5. Viterbi, A., “Error bounds for convolutional coding and asymptotically optimum decoding algorithm”, IEEE Trans. Inform. Theory, vol. 13, pp. 260-269, April 1967.